

20241008-Week5-黎明/My flight

Updated 1105 GMT+8 Oct 8, 2024

2024 fall, Complied by Hongfei Yan

已经进入第五周了，这标志着一个学期的时间已经过去了三分之一，真是光阴似箭。

机房位置安排-20241003.xls发布，第一次月考：2024.10.10 周四 7-8节。

考试时候只有 G++,Python: OJ上面 Python的时限是10倍（因为Python程序跑的慢），C/C++是没有放宽的。洛谷上面Python吃亏，因为没有放宽时限，拼不过C/C++。

2024fall 计概B 十月模考出题安排

题目难度暂定三档，参考CF, 800~900, 1000~1100, 1200~1400。

每个题目

- 1) 时间限制1000ms（如需要可加大），内存限制：65536KB（如需要可加大）。时间和内存如果需要修改，请注明。
- 2) 题面需要中英文双语。其中变量，给出范围。
- 3) 内容包括：描述、输入、输出、两组样例数据，及输出的必要解释。
- 4) 每个题目20组数据。另外，提供几组考虑边界或者特殊情况的手搓数据（可选）。
- 5) 给出题目类型tags和难度，例如：sortings, strings, implementation, 1100。
- 6) 可以是搬来的题目，如从 codeforces.com, leetcode.cn, luogu.com.cn搬来，注明出处。例如给出link，或者 Leetcode 1552。

2023fall 题目设计考虑

Updated 0934 GMT+8 Dec 25, 2023

1. 考察知识点：题目应涵盖语法、数据结构和算法等主要知识点，以测试学生对这些知识的理解和应用能力。
2. 难度控制：Easy题目大约需要10分钟完成；Medium题目大约需要20分钟完成；Tough题目因为每个学生的能力不同，因人而异。
3. 难度水平：题目的难度低于12份月考和平时最难的几次作业，以确保大多数学生都能够应对。
4. 简洁性：在题目设计时应避免设计过于繁琐的模拟题目，尽量保持简洁明了，突出重点。
5. 可解性：题目设计时应避免设计过于困难或者难以找到思路的题目，如高级贪心题目。学生应该能够有一定的思路来解决问题。
6. 知识点整合：可以将不同的知识点进行组合，例如结合数据结构、常规的贪心算法和动态规划等，以提高题目的综合性和难度。
7. 单一问题：除非特意增加难度，题目设计时尽量避免设计一题多问的情况，让学生集中精力解决一个问题。
8. 学生掌握情况：题目设计应根据班级大多数学生的掌握情况，确保题目的难度与学生的水平相匹配，既不过于简单也不过于困难。

9. 考试2小时：在考试前，学生应保持最佳状态，**充足的睡眠和身体健康非常重要**，这样才能发挥最佳水平。

一、Book 计算思维算法实践ch1&ch2

引自《计算思维算法实践》/Book_my_flight

前言

致使用本书的学生

当我们阅读时，另一个人替我们思考：我们只是重复他的思维过程。

When we read, another person thinks for us: we merely repeat his mental process.

——叔本华

计算思维离不了从数学的角度和计算机的角度考虑问题，有时候也涉及到对物理知识的应用。很多题目是对现实问题的再现、简化或变形，因此解题是训练新手解决现实问题的一种有效手段。在解题过程中，学习者需要保持清晰的思路缜密的逻辑，形成思考问题的方式方法，如此日积月累，自然而然就会养成计算思维。下面摘录两位初学者在培养计算思维的过程中给出的心得体会。

掌握了编程语言的语法后，学习的重点就转移到结合具体的题目体会解题中的思维方式。本书为练习题目选择了两个主要的编程平台，<https://codeforces.com/>（记为 CF），和 <http://openjudge.cn/>（记为 OJ）。

Codeforces 是国外著名的编程竞赛平台。题目种类丰富、分为不同的难度等级，还提供测试数据。很多题目非常有意思，有不少陷阱，很能训练思维。该平台最为友好的地方是代码和解题思路是公开的，能看到很多高水平的代码。题目页面右下角可以看到题目的 tags 和 tutorial 解题思路提示。如果想查看某个题目其他人提交的代码，替换链接中数字和最后一个字母，例如查看580C 可以访问

<http://codeforces.com/problemset/status/580/problem/C>。

Openjudge是北京大学研制的 ACM 训练和相关程序课程在线练习、考试系统，国内有很多高校和兴趣社区在其中开设了自己的小组，用户可以加到各个小组中做题。本书涉及到的中文题目都在 cs101 小组，所以文中出现的 OJ 通常指 <http://cs101.openjudge.cn>。OJ 上面的题目没有给出难度分类，因此本书特意给出了部分题目的难度分类作为示例，便于初学者从简单题目开始练习。在 OJ 上找到题目的方法是，访问 <http://cs101.openjudge.cn/>，登录后，点击“加入”按钮（表示加入“cs101”组，只需要初次访问时候加入 1 次），然后点击“题库（包括计概、数算题目）”。其中编号小于等于04152的题目是从“百练”小组 <http://bailian.openjudge.cn/> 引入到 cs101 小组的，因此题目 ID 尽量保持末四位一致。如“装箱问题”的题目 ID 是 01017，对应百练小组中的 1017 题目。

在这些平台上编程所用的语言主要选择 Python。个别题目超时不能 AC 时选择 C++。选用 Python 语言对于学习编程的人来说有个特别大的好处：有的题目用 C++ 写就直接 AC 了，但是用 Python 写会超时，这种情况下，学习者会琢磨如何优化，比如分析时间复杂度、更换算法等，这对于理解计算机相关原理特别有帮助。问题求解的关键在于算法，因此如何想到解决问题的办法是思维训练的重点，相比而言，语言之间的可移植性、代码的模块化程度和执行效率等方面的差异已经变得不那么重要了。

各个知识点对应的题目难度怎么定呢？本书采用多个平台横向比较，按照经验进行了对应。如图P-1所示，是国内比较知名的算法和程序设计类题库洛谷 (<http://luogu.com.cn/>)，其中的“普及/提高-”级别大致对应CF 中题目难度范围为“1200~1600”；“入门”级别大致对应CF中“800~1000”的范畴。学习者可以通过了解编程语法入门，而后通过有适配难度的题目进行思维训练，将解决问题的能力上升到更高的高度。

共计 6876 条结果

状态	题号	题目名称	显示算法	标签	难度	通过率
-	P1000	超级玛丽游戏			入门	
-	P1001	A+B Problem			入门	
-	P1002	[NOIP2002 普及组] 过河卒	NOIp 普及组	2002	普及-	
-	P1003	[NOIP2011 提高组] 铺地毯	NOIp 提高组	2011	普及-	
-	P1004	[NOIP2000 提高组] 方格取数	NOIp 提高组	2000	普及+/提高	
-	P1005	[NOIP2007 提高组] 矩阵取数游戏	NOIp 提高组	2007	提高+/省选-	
-	P1006	[NOIP2008 提高组] 传纸条	NOIp 提高组	2008	普及+/提高	
-	P1007	独木桥			普及-	
-	P1008	[NOIP1998 普及组] 三连击	NOIp 普及组	1998	普及-	
-	P1009	[NOIP1998 普及组] 阶乘之和	NOIp 普及组	1998	普及-	

图P-1. 洛谷网站的题目难度分级（截止 2021 年 12 月）

致使用本书的教师

本书的编写初衷是想将《计算概论》教学中积累的内容进行系统整理，使之成为以培养计算思维算法实践为目标的合适教材。因此本节在表P-1附上笔者 2021 年秋季学期《计算概论》课程安排，供阅读本书的教学工作者参考。第3章的附录 3A 初学编程常见问题，附录 3B 是课程作业和两位选课学生提供的答案。一学期有 16 周，每周 3 学时，练习到的题目约为 100~120 个。实践题目分布在 CF 和 OJ 两个编程平台上，在 OJ 平台的课程小组中有作者教学团队贡献的题目。

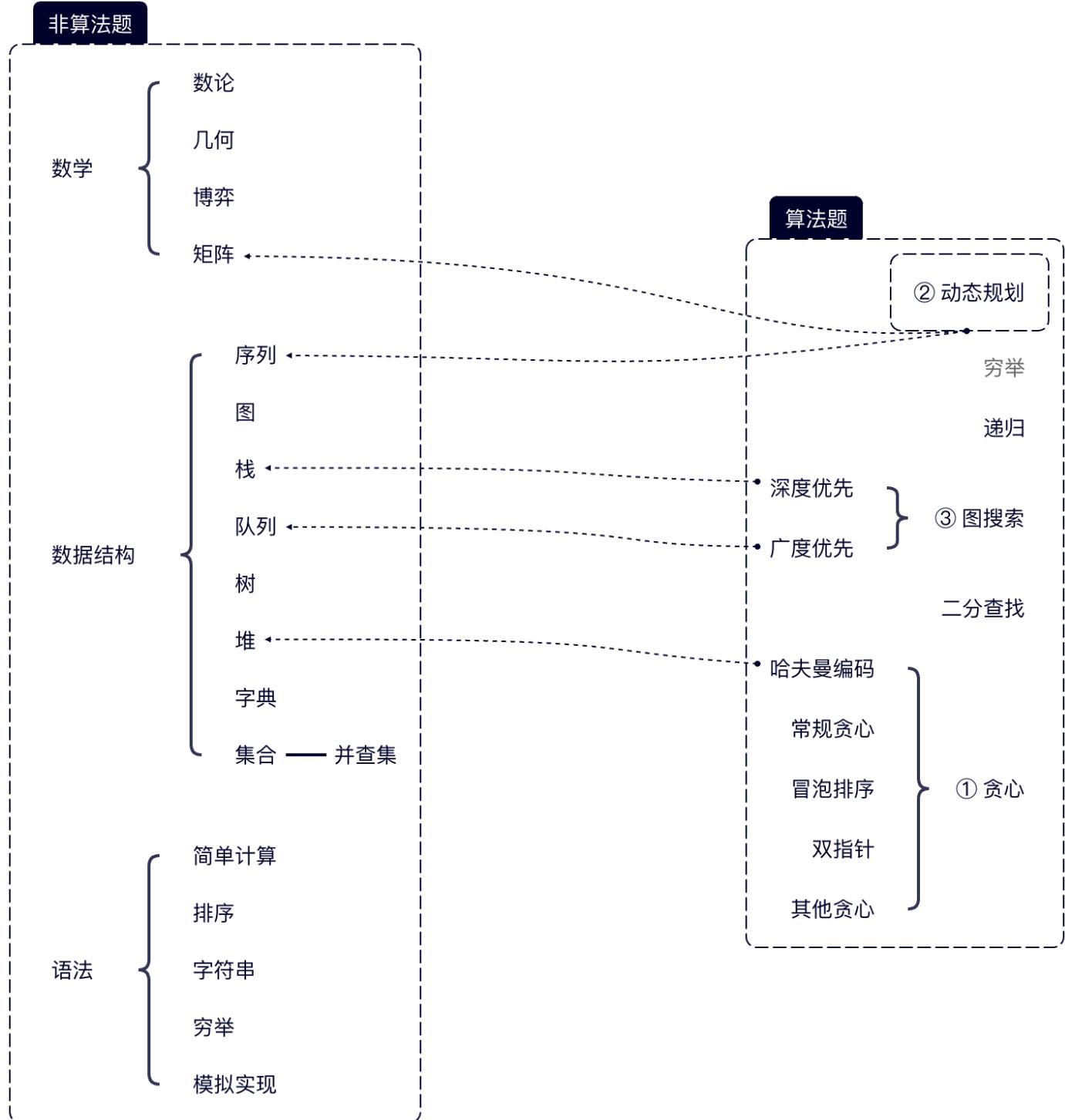
为了避免编程平台偶尔的访问故障、题目网址的变化，并确保本书的完整性，书中对题目给出了描述信息，及访问链接。所有题目均给出我们的题解以及简单明了的解题思路，下载地址是 <https://github.com/GMyhf/2020fall-CS101>。这些内容来自教师、助教，甚至选课学生的贡献，也包括从网上或者其他书籍整理借鉴的部分内容，对于这类内容，本书已尽可能详细地列出了参考来源，如有遗漏将尽量及时更正。

简明的解题思路在启发思考、激活联想中非常重要，如果能用一两句话说明的，就尽量避免冗长叙述。本着这一原则，本书有些题目的解题思路会甄选学习亲历者给出的言简意赅、具有点睛效果的解题思路。

表P-1 北京大学2021年秋季学期《计算概论》课程安排

1. 课程介绍和概述	2. Python语法
3. 计算机发展简史	4. 开发环境、程序语言
5. 月考、习题总结	6. 数据类型
7. 控制结构 (1/2)	8. 控制结构 (2/2)
9. 月考、二维数组	10. 递归、动态规划 (1/2)
11. 动态规划 (2/2)	12. DFS搜索
13. 月考、算法分析	14. BFS搜索
15. 课程总结、习题总结	16. 课时允许情况下的拓展项目

本书精心选择了难度在“普及/提高-”级别的120个题目，分门别类组织到如图P-2所示的编程题目类型框架中，供有需要的学习者和教学者参考。其中非算法类题目 69 个，算法类题目 51 个，如表P-2 所示。算法类题目重点学习①贪心、②动态规划和③图搜索，学习者可以参照这个划分选择自己先从哪里入手。



图P-2 编程题目类型框架

表P-2 每种题目类型需要练习的数目

题目类型	题目数
语法	44 (至少需要练习其中的 30 个)
数学	18 (其中 10 个是矩阵相关)
数据结构	7
贪心	18
动态规划	13
递归	3
二分查找	3
图搜索	14

编程题目中的非算法题是基础，侧重从计算机的工作模式来解决问题。算法题则通常既要了解计算机工作模式，又要结合数学的思考方法，并使用恰当的数据结构。算法是一步一步解决问题的过程，一个好的算法在时间或空间方面具有优势，不同类型的题目需要不同的算法来解决，同一个题目可能用不同的算法来解决。基本算法包括穷举（brute force）、贪心（greedy）、动态规划（Dynamic Programming, DP）、深度优先搜索（Depth First Search, DFS，是一种回溯法）、广度优先搜索（Breath First Search, BFS）、二分查找（binary search）和递归（recursive）。其中尤为重要的三类算法是：贪心、动态规划和回溯法。非算法题目包括：语法类、数学类和数据结构类。它们主要侧重语法的掌握以及对数据结构的熟悉和运用。因为 Python 等高级语言通常已将排序函数进行了很好的包装并提供调用，所以本章不单独列出排序算法一类，但冒泡排序稍有例外，它的本质是贪心法，因此在书中贪心算法一节会带出这个知识点。此外，虽然高级语言也经常会提供二分查找的现成函数，但是往往不能满足需求，多数还需要自己实现。博弈和并查集算法也是现实中常见的问题，本书把博弈归入了数学类，并查集归入了数据结构部分。加入这两种问题的相关算法也是为了拓展读者的解题思路。

计算思维算法实践学习建议：

1) 初学者可以从非算法题目中的语法部分开始，以熟练掌握编程语言的数据类型和两种控制结构为主要目标，对应3.2节。鉴于 CF 平台提供测试数据，所以推荐在 CF 上找难度为 800~1000 的题目开始练习，如果对英文题面有畏难情绪，也可以在 OJ 平台进行。语法题目通常涉及到简单计算、排序、字符串、穷举和模拟实现等类型。语法题目至少需要练习其中的 30 个，有基础或者进展快的读者可以相应减少本部分的练习数目。另外，穷举是很直观的算法，因此本书把穷举型题目归入了语法类；排序题目可直接使用 Python 编程语言提供的函数来实现就可以。

2) 数学类题目对应3.2节，主要是考察用数学知识解决计算问题，包括矩阵型、几何、数论和博弈等类型。本书所选的数学类题目难度级别对应中级或者难题。

① 其中，矩阵型题目是必须掌握的，因为很多算法的实现离不开对矩阵的理解。有时题干较长但题目不难，通常需要双重循环，行列要分清楚，另外注意行号、列号访问不要越界。需要练习 10 个矩阵题目。

② 几何题目需要能够推导出数学公式，或者找到现成的计算公式，如 CF270A: Fancy Fence。

③ 在稍微复杂一点的数论和博弈题目求解过程中，本书想特别谈一下超时的问题。首先，做题阶段遇到运行超时并不一定就是坏事。虽然编程历来要考虑计算效率，时间是程序可行性的一个重要考虑因素，但初学者遇到解题超时可以“倒逼”他们想解决办法。以数论题目 CF230B: T-primes 为例，这道题如果按常规的除余法来解，用 Python 编写的代码运行就容易超时，提交不通过。网上可以找到时间复杂度更低的筛法，如果学习者能不断设法优化，就

能既掌握时间复杂度原理，又体会到筛法的动态规划思想。这个题目如果用 C++ 来解，则失去了“倒逼”的意义，因为用简单的除余法就可以 AC，也就不会让人想到要改进筛法。

其次，遇到超时如何排查？通常是先考虑换用时间复杂度低的数据结构、算法，如果仍超时，再考虑使用 PyPy、C++ 或者 C 语言提交。以博弈题目 CF1425A: Arena of Greed 为例，这一题即使思路正确也容易超时。但如果提交时选择 PyPy 解释器来解释代码就可以 AC，这是因为 PyPy 的解释速度比较高，只是考虑兼容性问题，通常都是选择 Python。

3) 数据结构类题目对应3.4节，重点是分清楚每种数据结构（包括：字典、集合、序列、栈、队列、堆、图、树）的定义及时间复杂度，可以结合 2.6 节算法分析来理解。不少高级语言已经很好地封装了常用的数据结构，如 Python 中对字典、集合等数据类型的实现就已经在时间复杂度上做了优化考虑，可以方便的以数据类型方式直接使用。程序是数据结构加算法，算法题目的练习会大量结合数据结构使用，如深度优先搜索会用到栈，宽度优先搜索会用到队列。数组更是几乎每一个题目都要用到。鉴于算法部分会练习到数据结构，本章3.4节只安排了 7 个题目的练习。

4) 贪心 算法类包括 常规 和 非常规 两种，对应3.5节。常规的贪心算法类似于中规中矩的穷举法，根据题干描述容易确定贪心策略，只是需要注意逻辑清晰。读者可以结合 OJ01017: 装箱问题 来体会。非常规的贪心算法包括 双指针、冒泡排序 和 哈夫曼编码 等典型问题。这些题目的贪心策略不拘一格，不容易想到，需要先在纸上规划。有的属于难题级别，如双指针策略的 OJ022287: 田忌赛马。哈夫曼编码策略的 OJ18164: 剪绳子，如果使用堆这种数据结构，可以简化寻找最短绳子的过程。OJ12559: 最大最小整数 v0.3 一题对于初学也是难题，是冒泡排序使用了贪心策略。其他的题目如 OJ01852: Ants 涉及到物理知识，如果能想到弹性碰撞就比较容易找出解法。另外因为只是编程序而不需要证明，有时候想到一个贪心策略即使拿不准能够保证全局最优，也可以直接编程提交测试。本书中的贪心类题目练习达到 18 个。

5) 第3.6节进入具有挑战性的部分——动态规划。这类题目乍一看可以用 穷举法 来实现，但是会超时。可以先考虑更换 数据结构，比如把顺序访问的数组更换为时间复杂度低的字典或者集合。如果依然超时，还可以考虑采用类似于归纳法的动态规划，以空间换时间，对搜索过的路径或者子问题保存结果，避免之后遇到相同的子问题再重复计算。这类算法需要计算思维上的一个悟的过程，如果想不清楚可以在纸上或者利用 pythontutor.com 辅助规划。从相对简单的 序列型 动态规划题目开始，进而练习 矩阵型。本书中的动态规划题目练习有 13 个。

6) 递归 类题目对应3.7节，递归的特点是一个函数直接或间接地调用自己。所有的递归算法都有一个基础案例，递归算法必须改变其状态，并向着基础案例的方向发展。典型的递归题目是 OJ02694: 波兰表达式 和 OJ04147: 汉诺塔问题。本书有递归练习题目3个，递归法是回溯法的基础。

7) 二分查找 类的题目对应3.8节，二分查找是按照计算机的工作方式考虑问题的典型，即对于规模有限的问题，讲究先完成、再完美，用逻辑简单的“尝试”的思路来解决问题。虽然 Python 编程语言提供了二分查找函数，但是最好要自己实现二分查找来解决形形色色的实际问题。典型的二分查找题目是 OJ4135: 月度开销 和 OJ08210: 河中跳房子。本书中二分查找题目练习3个。

8) 图搜索 对应3.9节，DFS（深度优先搜索）和BFS（广度优先搜索）都是图搜索算法。它们的目标都是在图中寻找特定的节点或遍历整个图。DFS是一种通过沿着图的深度进行搜索的算法。它从起始节点开始，沿着一条路径一直深入直到无法再继续深入为止，然后回溯到前序节点，继续探索下一条路径。DFS通常使用递归或栈数据结构实现。BFS则是一种通过按照距离起始节点的距离逐层进行搜索的算法。它从起始节点开始，首先访问其所有的直接邻居节点，然后再逐层访问它们的邻居节点，直到找到目标节点或遍历完整个图。BFS通常使用队列数据结构实现。

回溯法可以看作是一种利用深度优先搜索策略的、在解空间树中搜索解的算法。它是一种通过不断地尝试可能的解决方案，并在不满足条件时进行回退的算法。它按深度优先搜索策略，从根结点出发深入搜索解空间树。当探索到某一结点时，要先判断该结点是否包含问题的解，如果包含，就从该结点出发继续探索，如果该结点不包含问题的解，则撤销该选择并回溯到上一级节点。若用回溯法的所有可能的搜索路径都搜索不到解，则问题的解不存在。

典型的深度优先搜索题目，如 OJ05585：晶矿的个数，OJ2754：八皇后。典型的宽度优先搜索题目，如 OJ21608：你和你比较熟悉的同学，CF580C：Kefa and Park。

DFS类的题目可以结合 动态规划、图、矩阵 等知识形成更难的题目，如 OJ01088：滑雪 和 OJ01661：Help Jimmy 就比较难；还可以结合剪枝技术，如 OJ04129：变换的迷宫。但是图搜索题目通常遵循一定的套路，一旦掌握，会感觉比 动态规划 题目简单。

本书中图搜索题目练习达到14个。

题目类型

为了感悟和提高计算思维，需要在简单、中等、困难三个层次的题目上，结合不同的算法或数据结构来考虑如何解决问题。本书配套的题目通过 标签 (tag) 来指明题目的类型，即主要考察的算法或数据结构，再按照不同难度层次给出相应题目的解答。

算法标签

沿用 Codeforces.com 习惯，每个题目主要考察的算法或者数据结构用tag标明，初学编程的人需要练习的题目，主要涉及到的tag包括：

bfs (breadth-first search, 宽度优先遍历，记为 bfs) , binary search, brute force,

dfs (depth-first search 深度优先遍历，记为 dfs) , data structures, dict , dp (dynamic programming, 动态规划，记为 dp) ,

graph, greedy, implementation, math, matrices, number theory, physics, recursion,

sorting, stack, strings, tree, two pointers。

在理解 bfs和dfs 之前，先介绍图遍历（graph traversal, also known as graph search）概念。图的遍历是指访问图中每个顶点的过程，这种遍历是按照访问顶点的顺序来分类的。一个问题的求解过程是从最开始的状态，利用已经存在的规则和条件改变当前状态，直到把当前状态变为最终目的状态。遍历是把中间出现的状态全部连接起来，形成一条遍历路径的过程。通过图的遍历，可以找到这条径。图的遍历算法主要有两种，一种是按照深度优先的顺序展开遍历的算法，也就是dfs；另一种是按照宽度优先的顺序展开遍历的算法，也就是bfs。bfs和dfs都是盲目的遍历方法，也就是说，遍历算法并不使用经验法则算法，并不考虑结果的可能地址，只是彻底地遍历整张图，直到找到结果为止。

数论（number theory）主要研究整数的性质。初等数论是用初等方法研究的数论，它的研究方法本质上说，就是利用整数环的整除性质，主要包括整除理论、同余理论、连分数理论。

我学算法的时候感觉也并没有多难。因为每一种算法都会有它独特的地方，即使最开始接触的时候不能理解，做几道例题并把题解看懂完全可以踏进新算法的大门。而学到最后，我发现同一个算法即使隐藏在不同的题目中，代码的核心部分也都极为相似。例如dp 的核心是状态转移方程，搜索的核心是用列表储存访问过的点，再将新点不断加入待访问列表中（不同搜索方法会以不同的顺序），理解了算法的核心操作那么算法就不再是一件难事了。【学习者 和沛淼，2021年秋】

难度分级

openjudge.cn 编程平台中题目没有难度指示，而codeforces.com 平台有提供题目难度指示，用整数表示，数字越大题目难度越大。针对零基础的初学者，本书把两个编程平台中需要练习的题目，按照简单、中等、困难 (Easy, Medium, Hard) 分为三个层次，每层进一步给出两级难度。难度依次升高的顺序是 Easy_Level1 -> Easy_Level2 -> Medium_Level1 -> Medium_Level2 -> Hard_Level1 -> Hard_Level2。针对每一级难度，给出相应题目的ID（OJ 开始的 ID 表示题目在 cs101.openjudge.cn 平台，CF 开始的 ID 表示题目在 codeforces.com 平台）、标题、tag。根据题目ID，可以在相应平台上找到题目。题目ID前面有*号的，表示该题目对于初学编程者较难。

书中给出了两个编程平台的部分题目的难度级别，而没有涵盖本书中所有题目。因为熟悉这些题目后，学习者就能够逐渐自行判断出其他题目的难度。两个平台都运行了多年，很多题目是经典题目，CF 平台本身提供答案、解题思路和测试数据，cs101.openjudge.cn 通常可以通过搜索引擎获得答案。鉴于 CF 提供信息多，编程开始推荐使用 CF 平台，练习题目过半时候，可以考虑开始并行使用 OJ 平台。

OJ 的题目列表如图P-3所示。在CF上的题目有一列表示难度，如图P-4所示主表格右数第二列的整数是难度级别指示。CF题目难度从800开始，800表示最简单的题目。800~1000难度的题目适合练习编程语法，1000~1600适合计算思维实践。有些读者对英文题目不太适应，可以先借助一些在线翻译工具帮助理解，题目读多了就会慢慢熟悉。



The screenshot shows the homepage of cs101.openjudge.cn. At the top, it says "Not Secure — cs101.openjudge.cn". Below that is a search bar with the placeholder "题目ID, 标题, 描述". To the right of the search bar are "登入" and "注册" buttons. The main content area has a large "CS101" logo with a "加入" button next to it. Below the logo is a section titled "»练习(247题)".

»练习(247题)						最新题目
题目ID	标题	通过率	通过人数	尝试人数	添加时间	
23554	小朋友春游	92%	23	25	2021-12-23	
23564	数论	78%	14	18	2021-12-23	
23556	小青蛙跳荷叶	93%	14	15	2021-12-23	
23558	有界的深度优先搜索	79%	22	28	2021-12-23	
23555	节省存储的矩阵乘法	93%	13	14	2021-12-23	

图P-3 cs101.openjudge.cn 题目列表，以中文为主、没有难度指示，不提供答案和解题思路

codeforces.com 2022/6/9 15:13

CODEFORCES
Sponsored by Telegram

HOME TOP CATALOG CONTESTS GYM PROBLEMSET GROUPS RATING EDU API CALENDAR HELP

MAIN ACMGURU | PROBLEMS SUBMIT STATUS STANDINGS CUSTOM TEST

#	Name			
1691F	K-Set Tree	combinatorics, dfs and similar, dp, math, trees		2500
1691E	Number of Groups	data structures, dfs and similar, dsu, graphs, greedy, sortings		2300
1691D	Max GEO Sum	binary search, constructive algorithms, data structures, divide and conquer, implementation, two pointers		1800
1691C	Sum of Substrings	brute force, constructive algorithms, greedy, math, strings		1400
1691B	Shoe Shuffling	constructive algorithms, greedy, implementation, two pointers		1000
1691A	Beat The Odds	brute force, greedy, math		800
1690G	Count the Trains	binary search, data structures, sortings		2000
1690F	Shifting String	graphs, math, number theory, strings		1700
1690E	Price Maximization	binary search, greedy, math, two pointers		1500
1690D	Black and White Stripe	implementation, two pointers		1000
1690C	Restoring the Duration of Tasks	data structures, greedy, implementation		800
1690B	Array Decrments	greedy, implementation		800
1690A	Print a Pedestal (Codeforces logo?)	constructive algorithms, greedy		800
1688C	Manipulating History	constructive algorithms, greedy, strings		1700
1688B	Patchouli's Magical Talisman	bitmasks, constructive algorithms, greedy, sortings		800
1688A	Cirno's Perfect Bitmasks Classroom	bitmasks, brute force		800
1687F	Koishi's Unconscious Permutation	fft, math		3500
1687E	Become Big For Me	combinatorics, constructive algorithms, greedy, math, number theory		3500

→ Pay attention
Before contest
Codeforces Round #798 (Div. 2)
33:21:13
Register now »
*has extra registration®

→ Filter Problems
Difficulty: —

→ Settings
 Show tags for unsolved problems
 Hide solved problems

→ Last unsolved
Name Last submission
1416A k-Amazing Numbers 96643658
1302I Deja vu 95570718
1061D TV Shows 46197973
200A Cinema 46136121
853A Planning 45497576
1070A Find a Number 44860459
271D Good Substrings 44658769

图P-4 codeforces.com 题目列表，以英文为主、提供答案、解题思路和测试数据

有的题目标题后面有 v0.x 字眼，说明题目有 bug，修改过。有的题目可以利用不同的算法或者数据结构求解，在 tag 中并列给出。初学者可以先不看有星号标识的题目，这些题目相对较难。

Easy_Level1

给出 4 个题目，Tags 包括：brute force, math, number theory, strings。

OJ02733：判断闰年, math

OJ02981：大整数加法, math/strings

OJ03143：验证“哥德巴赫猜想”, math

CF122A: Lucky Division, brute force, number theory, 1000

Easy_Level2

给出 9 个题目，Tags包括：binary search, brute force, greedy, implementation, matrices, math, number theory, strings, sorting。

OJ02808: 校门外的树, implementation

*OJ19943: 图的拉普拉斯矩阵, matrices

*OJ19944: 这一天星期几v0.2, math

OJ19949: 提取实体v0.2, strings

OJ03670: 计算鞍点, matrice

OJ04110: 圣诞老人的礼物, greedy

OJ18223: 24点, brute force/implementation

OJ21532: 数学密码, brute force/implementation/number theory

CF492B: Vanya and Lanterns, binary search/implementation/math/sortings, 1200

Medium_Level1

给出 17 个题目，Tags包括：dp, geometry, greedy, implementation, matrices, physics, recursion, stack, strings

OJ01852: Ants, physics/greedy

*OJ02694: 波兰表达式, recursion/strings/stack

*OJ02746: 约斯夫问题, implementation

OJ02757: 最长上升子序列, dp

OJ02760: 数字三角形, dp

OJ02806: 公共子序列, dp

OJ04015: 邮箱验证, strings

OJ03532: 最大上升子序列和, dp

OJ12558: 岛屿周长, matrice

OJ12560: 生存游戏, matrices

*OJ18106: 螺旋矩阵, matrice

OJ18161: 矩阵运算, matrices

OJ19942: 二维矩阵上的卷积运算v0.2, matrices

OJ21554: 排队做实验v0.2, greedy

CF270A: Fancy Fence, geometry/implementation/math, 1100

CF313B: Ilya and Queries, dp/implementation, 1100

CF545C: Woodcutters, dp/greedy, 1500

Medium_Level2

给出 15 个题目，Tags包括：dfs, dict, dp, greedy, implementation, number theory, stack , strings, two pointers

*OJ01017: 装箱问题, greedy

OJ02773: 采药, dp

OJ03704: 括号匹配, stack

OJ05585: 晶矿的个数, dfs

OJ12559: 最大最小整数v0.3, strings

*OJ12757: 阿尔法星人翻译官, implementation

OJ16528: 充实的寒假生活, greedy

OJ18108: 池塘数目, dfs

OJ18182: 打怪兽, dict

OJ18160: 最大连通域面积, dfs

OJ18211: 军备竞赛, greedy/two pointers

OJ23421: 小偷背包, dp

*CF189A: Cut Ribbon, brute force/dp, 1300

*CF230B: T-Primes, binary search/implementation/math/number theory, 1300

CF368B: Sereja and Suffixes, data structures/dp, 1100

Hard_Level1

给出 15 个题目，Tags包括：binary search, brute force, data structures, dfs, dp, implementation, matrices, two pointers

*OJ01088: 滑雪, dfs/dp

OJ02698: 八皇后问题解输出, dfs

*OJ02754: 八皇后, dfs

OJ02995: 登山, dp

*OJ04123: 马走日, dfs

*OJ04135: 月度开销, binary search

*OJ08210: 河中跳房子, binary search

*OJ16531: 上机考试, matrices

OJ19930: 寻宝, bfs

OJ19948: 因材施教, greedy

*OJ21577: 护林员盖房子, matrix/implementation

OJ21608: 你和你比较熟悉的同学, bfs/dfs

CF455A: Boredom, dp, 1500

CF466C: Number of Ways, binary search/brute force/data structures/dp/two pointers, 1700

CF1443C: The Delivery Dilemma, binary search/greedy/sortings, 1400

Hard_Level2

给出 6 个题目, Tags包括: bfs, dfs, dp, graph, greedy , tree

OJ01011: Sticks, dfs

*OJ01661: Help Jimmy, dfs/dp

*OJ02287: Tian Ji -- The Horse Racing, greedy

OJ04129: 变换的迷宫, bfs

*OJ18164: 剪绳子, greedy

*CF580C: Kefa and Park, bfs/dfs

编程题目逐行讲解

对于初次编程的同学, 如果看不懂代码, 可以参考编程题目逐行讲解视频, 如图P-5所示, 主要是2021年秋课
程助教录制的。这些有逐行代码讲解的视频, 是一学期中的作业题目, 放在 <https://space.bilibili.com/502167099/channel/collectiondetail?sid=454037>。

本书中涉及题目的代码是笔者整理的, 放在 <https://github.com/GMyhf/2020fall-cs101>。

《计算思维算法实践》练习题目逐行讲解。http://cs101.openjudge.cn

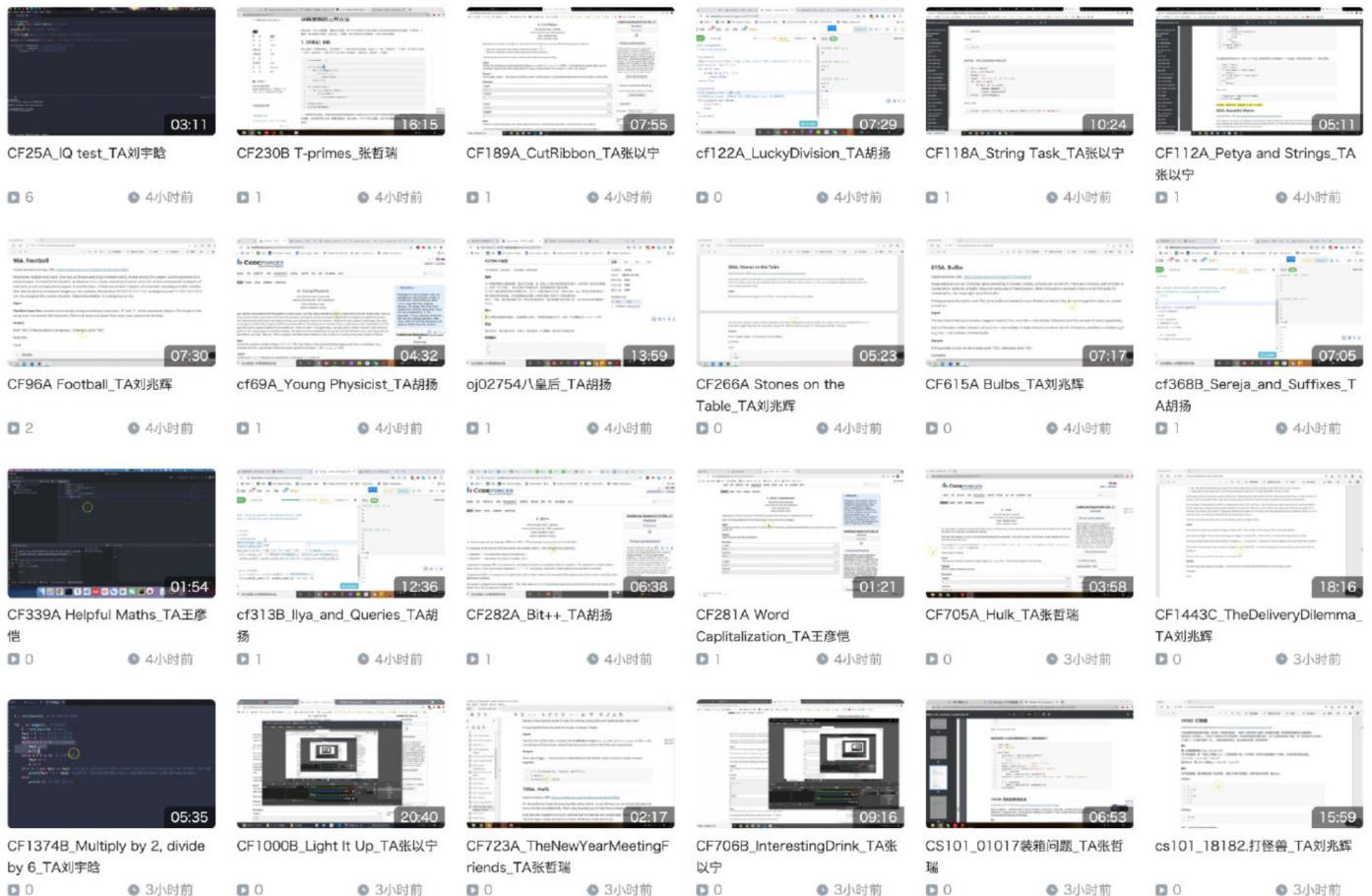


图 P-5 逐行讲解题目合集

第1章计算机文化历史和原理

1.1 计算机文化历史

1.1.1 计算机科学

在当今世界，几乎所有专业都与计算机息息相关。但是，只有某些特定职业和学科才会深入研究计算机本身的制造、编程和使用技术。用来诠释计算机学科内不同研究领域的各个学术名词的涵义不断发生变化，同时新学科也层出不穷。六个主要的计算机学科（discipline of computing）包括^{[1][2]}：

- 计算机工程学（Computer Engineering），是电子工程的一个分支，主要研究计算机软硬件和二者间的彼此联系。

- 计算机科学（Computer Science，记为 CS），是对计算机进行学术研究的传统称谓。主要研究计算技术和执行特定任务的高效算法。该门学科为我们解决确定一个问题在计算机领域内是否可解，如可解其效率如何，以及如何实现更加高效率的程序。时至今日，在计算机科学内已经派生了许多分支，每一个分支都针对不同类别的问题进行深入研究。
- 网络安全（Cybersecurity），涉及八个方面的安全，包括数据、软件、组件、连接、系统、人、组织和社会。
- 信息系统（Information Systems），研究计算机在一个广泛的有组织环境中的应用。
- 信息技术（Information Technology），指计算机相关的管理和维护。
- 软件工程学（Software Engineering），着重于研究开发高质量软件系统的方法学和实践方式，并试图压缩并预测开发成本及开发周期。
- 数据科学（Data Science），是一个新的计算领域，与数据分析和数据工程领域密切相关。它的一个定义是“一套指导从数据中提取知识的基本原则。并涉及通过（自动）分析数据来理解现象的原则、过程和技术”。

本书关注的是计算机科学。较大规模的致力于计算机科学的主要国际化组织有：美国计算机协会（Association of Computing Machinery，记为 ACM）；美国电气电子工程师协会（Institute of Electrical and Electronics Engineers，记为 IEEE）。

计算机科学是系统性研究信息与计算的理论基础以及它们在计算机系统中如何实现与应用的实用技术的学科。它通常被形容为对那些创造、描述以及转换信息的算法处理的系统研究。计算机科学包含很多分支领域；有些强调特定结果的计算，比如计算机图形学；而有些是探讨计算问题的性质，比如计算复杂性理论；还有一些领域专著于怎样实现计算，比如程序语言理论是研究描述计算的方法，而程序设计是应用特定的程序语言解决特定的计算问题，人机交互则是专注于怎样使计算机和计算变得有用、好用，以及随时随地为人所用。

现有的各种计算设备虽然可能在计算的时间、空间效率上有所差异，但在计算的能力上是等同的。尽管这个理论通常被认为是计算机科学的基础，可是科学家也研究其它种类的机器，如在已进入实际应用的并行计算机，以及目前主要还在理论设计阶段的概率计算机和量子计算机。在这个层面上来讲，计算机只是一种计算的工具。然而不要忘记著名的计算机科学家艾兹赫尔·韦伯·戴克斯特拉（荷兰语：Edsger Wybe Dijkstra）有一句名言：“计算机科学并不只是关于计算机，就像天文学不只是关于望远镜一样。”（"Computer science is no more about computers than astronomy is about telescopes."）

计算机科学根植于电子工程、数学和语言学，是科学、工程和艺术的结晶。它在 20 世纪最后的三十年间兴起成为一门独立的学科，并发展出自己的方法与术语。

早期，虽然英国的剑桥大学和其他大学已经开始教授计算机科学课程，但它只被视为数学或工程学的一个分支，并非独立的学科。剑桥大学声称有世界上第一个传授计算的资格。世界上第一个计算机科学系是在 1962 年由美国的普渡大学设立，第一个计算机学院于 1980 年在美国的东北大学设立。现在，多数大学都把计算机科学系列为独立的院系，还有些学校会根据学校的学科特色和师资队伍情况将它与工程系、应用数学系或其他学科联合。

计算机科学领域的最高荣誉是 ACM 设立的图灵奖，被誉为是计算机科学的诺贝尔奖。它的获得者都是某一领域最为出色的科学家和先驱。华人中首位图灵奖获得者是姚期智教授，他以对计算理论做出诸多根本性的、意义重大的贡献而获得 2000 年度的图灵奖。



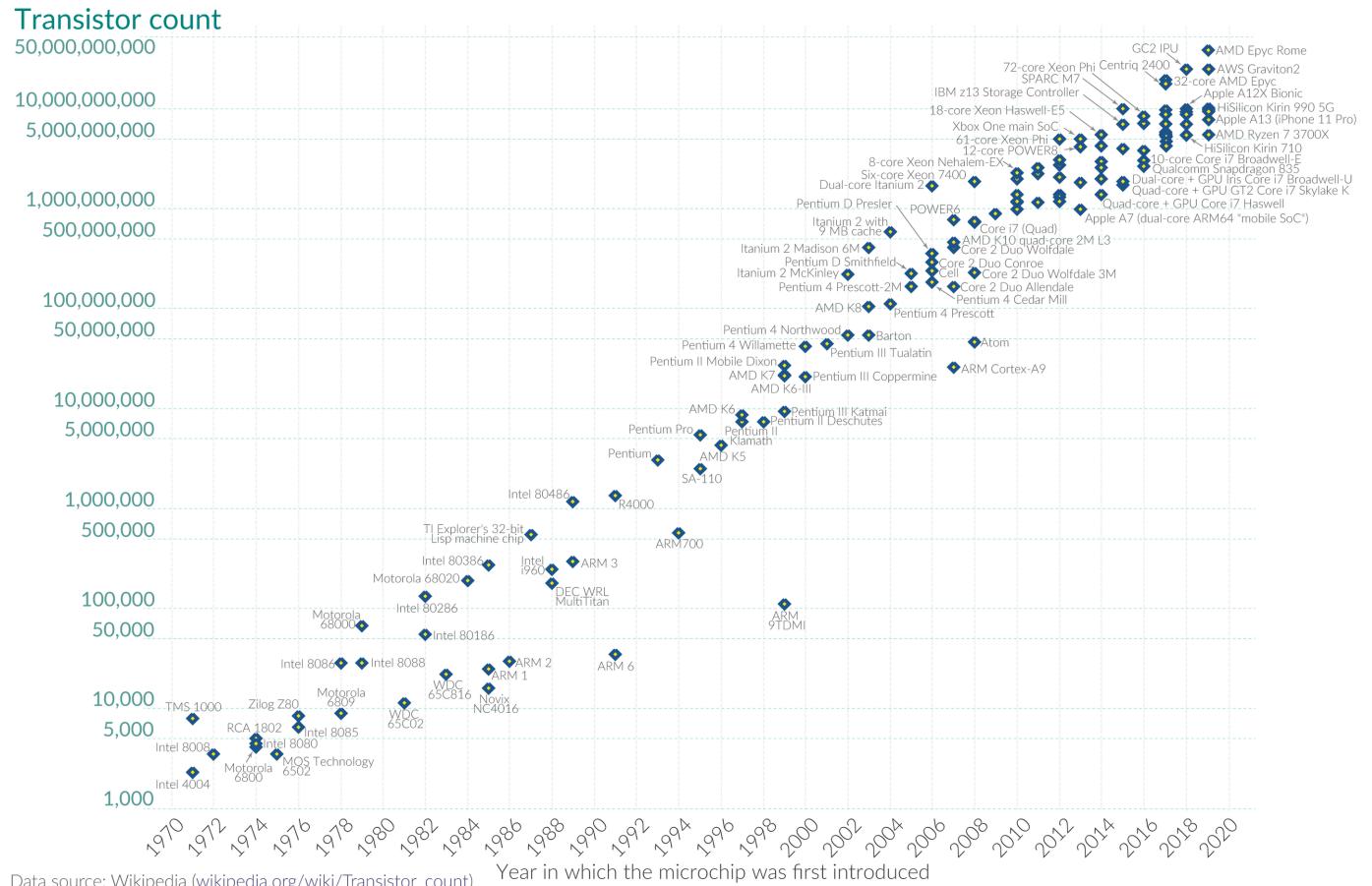
1.1.2 摩尔定律

摩尔定律 (Moore's law) 是由英特尔 (Intel) 创始人之一戈登·摩尔 (Gordon Moore) 提出的，被称为计算机第一定律。它是指集成电路上可容纳的晶体管数目，约每隔两年便会增加一倍。而后来被英特尔首席执行官大卫·豪斯 (David House) 修正为，预计18个月芯片的性能将会提高一倍，也就是说集成电路上晶体管集成度越高，将使造出的芯片的处理速度越快，性能大约以集成度的倍数增长。

Moore's Law: The number of transistors on microchips doubles every two years

Our World
in Data

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.



OurWorldInData.org – Research and data to make progress against the world's largest problems.

Licensed under CC-BY by the authors Hannah Ritchie and Max Roser.

图1-1 电脑处理器中晶体管数目的指数增长曲线符合摩尔定律 (A semi-log plot of transistor counts for microprocessors against dates of introduction, nearly doubling every two years) (注: 图片来源为 wikipedia.org, 2022年1月)

半导体行业大致按照摩尔定律发展了半个多世纪，对二十世纪后半叶的世界经济增长做出了贡献，并驱动了一系列科技创新、社会改革、生产效率的提高和经济增长。几乎所有衡量数字电子设备能力的标准都与摩尔定律有关：处理速度、内存容量，甚至数字相机的像素数量和大小。个人电脑、因特网、智能手机等技术改善和创新都离不开摩尔定律的延续。所有这些也都在以大致指数级的速度提升。这极大地提高了数字电子产品在世界经济几乎每一个领域的实用性。摩尔定律描述了 20 世纪末和 21 世纪初技术和社会变革的驱动力。

戈登·摩尔是CPU 生产商 Intel 公司的创始人之一。1965 年提出“摩尔定律”，1968 年创办 Intel 公司。摩尔 1929 年 1 月 3 日出生在美国加州的旧金山，曾获得加州大学伯克利分校的化学学士学位，并且在加州理工大学 (CIT) 获得物理和化学两个博士学位。50 年代中期他和集成电路的发明者罗伯特·诺伊斯 (Robert Noyce, 1927 年 12 月 12 日 – 1990 年 6 月 3 日) 一起，在威廉·肖克利半导体公司工作。后来，诺伊斯和摩尔等 8 人集体辞职创办了半导体工业史上有名的仙童半导体公司 (Fairchild Semiconductor)。仙童成为现在的 Intel 和 AMD 之父。1968 年，摩尔和诺伊斯一起退出仙童公司，创办了 Intel。Intel 初期致力于开发当时计算机工业尚未开发的数据存储领域，后来，Intel 进行战略转移，专攻微型计算机的核心部件 CPU。

1.1.3、1.1.4 skip

1.1.5 计算机发展简史

硬件是计算机中固态可见的部分，看得见，摸得到。计算机械有着很长的历史，然而最初的计算设备是为特殊用途而设计的。大约公元前 100 年出现的安提凯希拉装置 (Antikythera mechanism)，1901 年于希腊安提凯希拉岛上的一艘古船残骸中被发现的随船沉没的钟形装置。该装置由一系列可以被控制的铜质齿轮（超过 30 个）、曲柄和刻度盘组成。百余年后，科学家终于在现代先进技术辅助下，揭开它惊人谜底——这座名为“安提凯希拉装置”的装置竟然是一台两千年前的超级天文“计算机”，天文学家计算天体运行周期的工具，它的先进性在其制建成后千年间无人超越。算盘之类的算术工具，尤其是在亚洲地区已经有上千年的使用历史。在约翰·纳皮耶描述了对数之后不久，计算尺于 17 世纪早期被发明了出来。和现代计算机最接近的先驱是雅各提花织机，在 19 世纪左右由法国的约瑟夫·马里·雅各发明，通过多行打孔的矩形卡片来确定特定的编织图案。这种提花织机是“可编程的”，可以通过改变打孔卡片发出的指令来改变编织的图案。这种可节省人力的纺织机械的发明，导致很多编织工人失去了工作，并因此造成了社会混乱。1811~1816 年在英国发生的卢德运动就是一个针对机械化的暴力抗议。现代的计算机技术同样会对一些传统工作带来冲击。

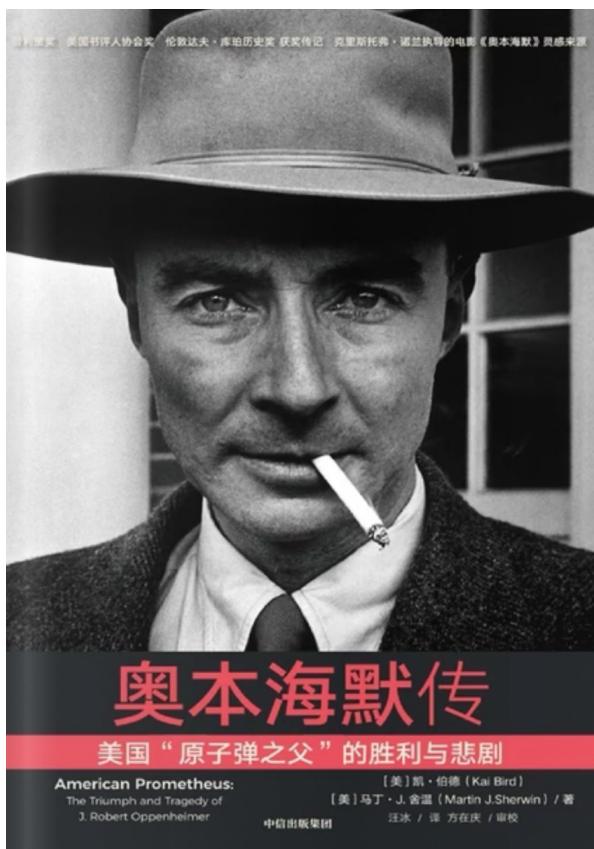
今天的计算机来源于 20 世纪中期的英国，从查尔斯·巴贝奇 (Charles Babbage, 1791 年 12 月 26 日 – 1871 年 10 月 18 日) 的作品发展而来。巴贝奇是一个对导航和天文学有兴趣的科学家，这两个领域都需要有数值表以计算位置。他的毕生心血花在建筑计算机械上，希望能建造一个可以将枯燥易错的手工计算过程机械化的机器。由于很多原因，未能实现这个雄心壮志，其中一个原因是财政赞助人。即使这样，他的设计还是很完美的。借助那个时代的材料和工具，他的一些机器在现代得以实现。在伦敦的科学博物馆可以看见差分机。**“差分”的含义是把函数表的复杂算式转化为差分运算，用简单的加法代替平方运算。**差分机使用有限差分方法来机器计算多项式函数的值。有限差分方法是个简单但功能强大的技巧，它用重复加减的过程来避免需要的乘法和除法。

查尔斯·巴贝奇激励了奥古斯塔·艾达·拜伦对数学和他的计算设备的兴趣。艾达是诗人乔治·拜伦的女儿。她被认为是世界上第一个程序员，Ada 编程语言也以她的名字命名。

19 世纪晚期，赫尔曼·何乐礼 (Herman Hollerith Herman Hollerith, 1860 年 2 月 29 日 – 1929 年 11 月 17 日) 为美国人口普查局工作，他设计并制造出了和人工操作相比可以更快地将普查数据列成表格的机器。采用雅各提花机的灵感，何乐礼通过在卡纸上打孔以将普查数据编码成可以利用他的机器处理的格式。1880 年的普查结果利用 8 年时间才全部录入表格，而借助何乐礼的打孔卡片和打孔卡片制表机 (Tabulation Machine)，1890 年的

普查结果只用了一年时间便得以完成。何乐礼在 1896 年，创办了制表机器公司（Tabulating Machine Company），它是 IBM 的前身。

被誉为自然科学史上的三大计划是阿波罗登月计划、曼哈顿原子弹计划、和人类基因组计划。**1942 年开始的美国的原子弹计划核爆计算促成了计算机的诞生**。第一代（1946~1958 年）计算机，其主要逻辑元件是电子管。电子管是和圆柱形电灯泡大小形状类似的电子设备，它们不仅贵，而且笨重、脆弱，能耗也很高。第一代计算机体积很大，通常要占用好几个房间，运算速度为每秒几千次。这个时候的计算机主要用于科学计算。第一个完全由电器组成的计算机是 ENIAC（Electronic Numerical Integrator and Computer，电子数值积分器计算机）于 20 世纪 40 年代由宾夕法尼亚大学的埃克特（Presper Eckert）和莫克利（John Mauchly）于费城建造，它占据了一个很大的房间，并且需要大量的电力支持。它可以每秒做 5,000 次加法。ENIAC 最初计划用于弹道计算等领域，但是直到 1946 年，二战结束很久之后才建造出来。计算设备可以将操作指令和数据以相同的方式存储，但是 ENIAC 并没有将指令和数据都保存在内存里。第一个真正可以将程序和数据存储在一起的计算机 EDSAC（Electronic Delay Storage Automatic Calculator，电子延时存储自动计算器）于 1949 年在英国制造。



1961 年到 1972 年的阿波罗登月计划，参加的有两万家企业、200 多所大学、80 多个科研机构，总人数超过 40 万人。由于阿波罗计划的出现，导致了上世纪 60~70 年代 20 多年包括火箭、雷达、无线电制导、合成材料、计算机等一大批高科技工业群体的诞生。所有这些技术又应用到民用方面，带动了整个科技发展和工业的繁荣。

1947 年晶体管的发明，以及 1958 年集成电路的发明，现代计算机时代正式开始。这些技术使电子系统有可能变得更小，更便宜，同时也更快。第二代（1958~1964 年）计算机，其主要逻辑元件是晶体管，运算速度可达每秒几十万次。出现了高级程序设计语言，极大地简化了编程工作，应用领域也增至数据处理。

随着时间的推移，集成电路的研究发展壮大了硅谷的芯片产业，计算机时代也随着英特尔 1971 年推出了世界上第一款微处理器（CPU）4004 拉开序幕。CPU 将计算机的能力带到了新的高度，创造了革命性的生产生活方式。第三代（1964~1971 年）计算机，其主要逻辑元件是中小规模集成电路，运算速度达每秒几十万次到几百万 (10^6) 次。操作系统、高级程序设计语言、编译系统等基本软件在这一时期初步成型。计算机应用到各个领域。

第四代（1971~至今）计算机，主要逻辑元件是大规模和超大规模集成电路，运算速度达到了每秒上亿次，甚至上千万亿（ 10^{15} ）次，操作系统不断完善；微型机在家庭得到了普及，并开始了计算机网络时代。

1.2 计算机原理

硬件和软件的组合形成可用的计算系统。硬件通常由软件指示以执行任何命令或指令。计算机硬件包括计算机的物理，有形部件或组件，例如主板（Main Board）、中央处理器（Central Processing Unit, CPU）、显示器、键盘和鼠标。计算机软件包括系统软件和应用软件。系统软件负责管理计算机系统中各种独立的硬件，使得它们可以协调工作。系统软件中常用的操作系统有 Linux、macOS、Unix、Windows 等。应用软件是为了某种特定的用途而被开发的软件。常见的应用软件有文字处理、程序设计、网页浏览器、输入法和媒体播放器等。

本节先对主要的计算机硬件进行感性认识，然后介绍本书认为对计算机设计具有重要意义的三个原理：图灵机、进程的虚拟地址空间和 ASCII 表。因为课时有限，练习题目偏多，其他涉及到的原理知识，会在题解中有体现，比如：时间复杂度，深度优先搜索和宽度优先搜索哪个快，浅拷贝和深拷贝，全局变量和局部变量，二进制位操作等。这样一个好处是大家在接触了与计算机相关的最重要的硬件和原理后，在不过度纠结于细节的情况下，可以更早的接触编程语言，开始计算思维的编程实践，在实践中加深对计算机原理的理解。

1.2.1 对计算机的感性认识

为了让零基础的同学，产生感性认识，先给出使用 Windows 操作系统的计算机和部件，然后展示使用 macOS 系统的 Mac mini 和 MacBook Pro 笔记本。

张以宁助教在2021年9月录制了自己组装的计算机介绍，如图1-2所示是整机外观，21分钟视频讲解在 http://www.bilibili.com/video/BV1D5411D7by?spm_id_from=333.999.0.0。



图1-2 一台助教同学 DIY 组装计算机的外观

表1-1 2021年装机选型方案

配件	型号	官方价(京东)
CPU	AMD 锐龙Ryzen7 5800X	¥ 2269
主板	华硕 TUF GAMING B550M-PLUS (WI-FI) 重炮手	¥ 799
散热器	九州风神 堡垒360V2 一体式水冷	¥ 699
内存	影驰 星曜16G*2 DDR4 3200	¥ 1038
显卡	影驰 RTX3070Ti 星耀 OC	¥ 6499 (?)
硬盘	西数 WD_BLACK SN850 1T SSD固态	¥ 1499
机箱	影驰 超新星 绝白版	¥ 339
电源	鑫谷 GP850G 750W金牌全模组	¥ 499

官方价总计：¥ 13641

选型说明：受比特币价位猛涨（“狂潮”）和芯片产能不足的双重影响，2021年的显卡市场价格非常魔幻，各厂商的显卡首发价都成了一个“理论数字”。例如表1-1中的显卡影驰星曜 RTX3070Ti，首发价 6499，实际价位可能在 7500 以上，还严重缺货。在这种形势下反而通过购买整机，其实就是搭购其他配件能获得较高的性价比。2021年8月份在价位相对较低的时候整机购入，实际价格¥ 12046。

一些其他装机选型知识：

- 1) CPU 的成产厂商主要是 Intel 和 AMD 两家做，他们的 CPU 底部脚位设计不同。主板厂商有多个，如华硕、微星、技嘉等，要按照 CPU 来选择适配的主板。除了分别购买CPU和主板，也可以在网上买到“板U套装”，其价格通常比单买便宜一些。
- 2) 关于显卡。与 CPU 不同，一个完整的独立显卡除了核心的处理器之外，还要兼顾显存、供电、散热等设计。英伟达(NVIDIA) 和 AMD 自己会设计一套完整的产品，称为“公版”(Founders Edition)，如图1-3所示。然而公版卡的产量一般很少，更多的是，英伟达和AMD 会把核心的显卡处理器和显存卖给其他硬件厂商，再由各厂商自己做供电、散热，当然还有外观等设计，这种称为非公版卡，如图1-4所示，也是市面上能买到的绝大部分显卡。同一型号的显卡，各厂家还可能在供电等设计上“用料”质量不同，推出从低端到高端多个产品，市场上俗称“丐版卡”和“旗舰卡”。2021年9月英伟达推出的显卡为30系列，性能从低到高为 3060, 3060Ti, 3070, 3070Ti, 3080, 3080Ti, 3090。AMD 最新为 RX6000 系列，性能从低到高为 6600XT, 6700XT, 6800XT, 6900XT。



图1-3 3060Ti 公版卡外观（注：图片来源为 mydrivers.com，2020年12月）

